An exact algebraic ϵ-constraint method for bi-objective linear integer programming based on test sets

María Isabel Hartillo-Hermoso [a, 1], Haydee Jiménez-Tafur [b], José María Ucha-Enríquez [a, b, 2, *]

[a] Departamento de Matemática Aplicada I, E. T. S. de Ingeniería Informática, Universidad de Sevilla, Av. Reina Mercedes s/n, Sevilla 41012, Spain
[b] Instituto de Matemáticas, Universidad de Sevilla, Avda. Reina Mercedes, s/n, Sevilla 41012, Spain

Keywords:
Multiple objective programming
Non-dominated set
Pareto set
ϵ-constraint method
Unbounded Knapsack Problem
Algebraic test sets

ABSTRACT

A new exact algorithm for bi-objective linear integer problems is presented, based on the classic ϵ-constraint method and algebraic test sets for single-objective linear integer problems. Our method provides the complete Pareto frontier $\mathcal{N}$ of non-dominated points and, for this purpose, it considers exactly $|\mathcal{N}|$ single-objective problems by using reduction with test sets instead of solving with an optimizer. Although we use Gröbner bases for the computation of test sets, which may provoke a bottleneck in principle, the computational results are shown to be promising, especially for unbounded knapsack problems, for which any usual branch-and-cut strategy could be much more expensive. Nevertheless, this algorithm can be considered as a potentially faster alternative to IP-based methods when test sets are available.

1. Introduction

In real-life problems, the pursued objectives are multiple and decision makers usually need complete knowledge of the best decisions to be made from those different points of view. The concept of *non-dominated solution or Pareto optimal* gathers the solutions that cannot be improved in any objective without degrading at least one of the other objectives.

Multi-objective Integer Programming (MOIP) is the branch that deals with this kind of problems in the case of integer variables. Generation methods compute the whole space of Pareto optimal solutions. Among these type of methods, the *weighted-sum method* transforms a multi-objective problem into several single-objective problems that arise from the linear combination of the different costs. If the Pareto set of points is convex this method provides the whole set of non-dominated solutions, but only the supported Pareto optimal points are obtained in the general case. On the other hand, the ϵ-*constraint technique* generates a grid in the objective space with ranges between the costs of ideal and nadir points and, for each point in the upper-bound set (cf. Ehrgott and Gandibleux (2007) and Bringmann, Friedrich, Igel, and Voß (2013)), a single-objective IP is solved, thereby avoiding incremental movements through the grid.

In Laumanns, Thiele, and Zitzler (2006), an adaptive variation of the constraints that depends on the Pareto-optimal solutions found so far has been proposed. They use a lexicographic optimization with respect to the costs in the constraints in order to avoid weakly efficient Pareto optimal solutions. In their process slackness of the constraints are forced. The time complexity of their algorithm is $\mathcal{O}(|\mathcal{N}|^{p-1})$, where $|\mathcal{N}|$ is the number of Pareto-optimal solutions and p the number of objectives.

In Ehrgott and Ruzika (2008), the main weaknesses of the ϵ-constraint technique are pointed out: the lack of easy-to-check conditions for properly efficient solutions and the inflexibility of the constraints. In order to address these weaknesses, slack and surplus variables are added.

In Mavrotas (2009), the so-called *augmented* ϵ-*constraint method* is presented. By using the previous idea of slack variables, an effective implementation of the ϵ-constraint technique is given. In seeking non-dominated solutions, the algorithm changes the cost function, by rewarding whenever the slack variables are positive. This method is improved in Mavrotas and Florios (2013), and a variation is proposed in Zhang and Reimann (2014).

In Kirlik and Sayın (2014) a new method is described, which is also based on the ϵ-constraint technique, with an innovative way of partitioning the search space in terms of rectangles. This technique presents a two-stage formulation for each problem in contrast to the lexicographic option.

[*] Corresponding author.
E-mail address: ucha@us.es (J.M. Ucha-Enríquez).

[1] Partially supported by MTM2016-74983-C2-1-R and MTM2016-75024-P.

[2] Partially supported by MTM2016-74983-C2-1-R, MTM2016-75024-P and P12-FQM-2696.

Other alternative approaches have been proposed, such as the triangle-splitting method in Boland, Charkhgard, and Savelsbergh (2014) for the more general case of mixed linear integer programming, and methods based in advanced branch-and-bound analysis (cf. Ralphs, Saltzman, & Wiecek (2006)).

Two additional algebraic approaches to MOIP have been presented. In Blanco and Puerto (2009) the construction of the so-called *partial Gröbner bases* (that turns out to be similar to *Graver bases* in many cases) is proposed. In Blanco and Puerto (2011) the multiobjective problem is reduced to finding solutions of a system of polynomial equations induced by optimality conditions: the necessary Karush-Kuhn-Tucker, the Fritz-John and the multiobjective Fritz-John nondominance conditions. This second approach generalises (Bertsimas, Perakis, & Tayur, 2000), in which single objective optimisation problems are treated applying Gröbner bases for solving systems of polynomial equations.

We present a new exact algebraic method to solve bi-objective linear integer problems. Our first contribution is theoretical: we provide a new way of understanding the classical ϵ-constraint technique under a new perspective. Our approach uses the so-called *test sets* associated with single-objective integer programming problems, and takes advantage of their special characteristics. A test set is a set of directions that guides the movement from any feasible point until the optimum of a linear integer programming problem is reached. Once a test set for a linear integer problem is obtained, it can be solved by reducing any feasible point with the elements of the test set instead of passing it to an optimizer. The reduction is the simple iterative process, in which any element of the test set that produces a feasible point can be used repeatedly. Test sets can be computed in several ways. It is proved in Thomas (1995) that *Gröbner bases* provide the minimal test set for a fixed total ordering that is compatible with the linear cost function of the program considered. Test sets do not depend on the right-hand side (RHS) of the constraints. Interested readers can consult references (Sturmfels, 1996) and (Bertsimas & Weismantel, 2005) for further details on this topic. Although the computation of Gröbner bases is a hard task in general, since it is highly sensitive to the number of variables (the degree of the polynomials in the bases in the worst complexity case is $\mathcal{O}(kd^{2n})$ for ideals generated by k polynomials in n variables of degree at most d, see Mayr and Meyer (1982), in our experiments the algorithm is fairly competitive in the *unbounded* knapsack problem. Its performance in the 0–1 knapsack problem remains unexceptional because each 0–1 variable adds one equation and one variable in this context. Gröbner bases can sometimes be obtained theoretically, thereby avoiding the potential disadvantages of the bare computation. This is the case of the family presented in Laumanns et al. (2006). In Appendix A, we provide the technical details of how to obtain the closed formula of the Gröbner bases that are required for this family to be treated with our method for any number of variables.

The method introduced in this paper, in addition to its theoretical contribution, provide some potential advantages to apply the ϵ-constraint framework. To begin with, our method detects accurately the efficient solutions. Redundant cases are circumvented: test sets point out exactly which values of the slack variable (of the added constraint due to the first cost) will produce new efficient solutions. Secondly, to solve the IPs we do not pass the problem to a solver, but use a process called reduction to obtain the optimal solutions and it may be considered therefore as an alternative to IP-based methods. Thirdly, our method uses only one test set to solve all the single-objective problems, because the same test set is valid to solve a given problem in spite of the RHS of its equations. Finally, we avoid weakly efficient solutions. We show how test sets, computed with Gröbner bases with respect to a suitable total ordering, solve in one step each problem into a ϵ-constraint framework, obtaining directly efficient solutions. In contrast, the algorithm of Laumanns et al. (2006) depends on the number of problems to be solved by the optimizer in order to manage a lexicographic setting.

The remainder of the document is structured as follows. In Section 2 we introduce the problem and the tools we are going to use. In Section 3 we present the theorems in which our algorithm is based. In Section 4 we treat in detail a worked example to help the understanding of our method. In Section 5 we present some computational experiments to test the performance of our algorithm. In Section 6 we show the conclusions and, finally, in the Appendix we show an example of how the Gröbner basis of a given problem can be sometimes obtained theoretically, avoiding its potentially expensive calculation with a computer.

2. Preliminaries

Our interest lies in solving the bi-objective linear integer optimization problem in its standard form, that is,

$$\begin{array}{ll} \min & c_1(\mathbf{x}), c_2(\mathbf{x}) \\ \text{s.t.} & A\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \in \mathbb{Z}^n_{\geq 0} \end{array} \tag{1}$$

for $A \in \mathbb{Z}^{m\times n}$, $\operatorname{rank}(A) = m$, $\mathbf{b} \in \mathbb{Z}^m$ and c_1, c_2 linear functions with integer coefficients. Since in general there is no feasible point that minimises both cost functions, we are concerned about the *efficient points*: those feasible points $\mathbf{x}^\star$ such that there is no feasible $\mathbf{x}'$ with $c_k(\mathbf{x}') \leq c_k(\mathbf{x}^\star)$ with at least one strict inequality for $k = 1, 2$. If $\mathbf{x}^\star$ is an efficient point, $(c_1(\mathbf{x}^\star), c_2(\mathbf{x}^\star))$ is a *non-dominated (or Pareto) point* in the objective space. If we replace the condition $c_k(\mathbf{x}') \leq c_k(\mathbf{x}^\star)$ with $c_k(\mathbf{x}') < c_k(\mathbf{x}^\star)$, then we obtain *weakly efficient (or weak Pareto)* points. We will denote $\mathcal{X}$ as the set of efficient points and $\mathcal{N}$ as the set of non-dominated points, the *Pareto frontier*.

We will assume that the feasible region for problem (1) is finite, and therefore the Pareto frontier $\mathcal{N}$ is also finite. In this work, we show how to obtain a set $\mathcal{X}^\star \subset \mathcal{X}$ that is a *minimal complete set of efficient points* (that is, if $\mathbf{x}^a, \mathbf{x}^b \in \mathcal{X}^\star$ then $(c_1(\mathbf{x}^a), c_2(\mathbf{x}^a)) \neq (c_1(\mathbf{x}^b), c_2(\mathbf{x}^b))$ and $|\mathcal{X}^\star| = |\mathcal{N}|$, as in Ehrgott (2005)).

The ϵ-constraint technique, a common method to address problem (1), solves many problems of the form

$$\begin{array}{ll} \min & c_k(\mathbf{x}) \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} \\ & c_j(\mathbf{x}) \leq \epsilon_j, \quad j = 1, 2 \ (j \neq k) \\ & \mathbf{x} \in \mathbb{Z}^n_{\geq 0} \end{array} \tag{2}$$

for fixed $k = 1, 2$ and suitable values of ϵ_j in order to find efficient points. More precisely, the efficiency of a point $\mathbf{x}^\star$ is related to the fact that $\mathbf{x}^\star$ is an optimal solution of problems of type (2) for $k = 1, 2$, as the following theorem of Ehrgott (2005) states for the bi-objective linear case:

Theorem 2.1. *A feasible solution $\mathbf{x}^\star$ of a linear MOIP is efficient if and only if there exists $(\epsilon_2, \epsilon_1) \in \mathbb{R}^2$ such that $\mathbf{x}^\star$ is an optimal solution of the corresponding problems (2) for $k = 1, 2$.*

Consequently, we have a family of IPs for which the RHS varies for different ϵ_j. This fact provides us the opportunity to consider an algebraic tool to deal with integer problems (IPs): the *test sets*. Given an IP in standard form (no inequalities)

$$\begin{array}{ll} \min & c(\mathbf{x}) \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \in \mathbb{Z}^n_{\geq 0} \end{array} \tag{3}$$

for $A \in \mathbb{Z}^{m\times n}$, $\operatorname{rank}(A) = m$, $\mathbf{b} \in \mathbb{Z}^m$ and assuming c as a linear function with coefficients in $\mathbb{Z}^n$, we have the following result (cf. Schrijver (1986)):

Theorem 2.2. *Given problem (3) there exists a set $\mathcal{T} \subset \{\mathbf{t} \in \mathbb{Z}^n : A\mathbf{t} = 0\}$ valid for any RHS, with the following properties:*

1. *For any feasible solution* $\mathbf{x}$ *of* (3) *that is not optimal, there exists* $\mathbf{t} \in \mathcal{T}$ *such that* $\mathbf{x} - \mathbf{t}$ *is feasible and* $c(\mathbf{x} - \mathbf{t}) < c(\mathbf{x})$.
2. *Given the optimal solution* $\mathbf{x}^\star$ *of* (3), $\mathbf{x} - \mathbf{t}$ *is not feasible for any* $\mathbf{t} \in \mathcal{T}$.

Any set satisfying these conditions is called a *test set* for (3). The existence of test sets for an IP implies a straightforward algorithm to find its optimum: we start from a feasible point and subtract (we will say *reduce* too, because of the correspondence between points and monomials, and between test sets and *Gröbner bases*) elements of the test set on the condition that feasible points are obtained.

Several alternatives are available for the computation of test sets. It is proved in Thomas (1995) that Gröbner bases, computed with respect to a fixed total ordering $\prec_c$ *compatible with the linear cost function* of the considered program, provide a (minimal) test set for (3). Therefore this test set is *valid for any RHS of the constraints.*

In general, the computation of Gröbner bases requires a total ordering (cf. Cox, Little, and O'Shea (2005) or Sturmfels (1996)). To solve a linear program, the total ordering $\prec_c$ must be compatible with the cost function c: the selected total ordering needs to compare points using the cost c in the first place and then break ties with another total ordering. This second total ordering can be another cost function or a lexicographic ordering $<_{\text{lex}}$ for which $\mathbf{a} <_{\text{lex}} \mathbf{b}$ when the first non-zero entry of $\mathbf{b} - \mathbf{a}$ is positive. These test sets produce optimal solutions of a given IP that are *unique*: they have the smallest value of c and they are the best solutions with respect to $\prec_c$.

By means of using test sets, only *one* test set needs to be computed for *all* the problems that have to be solved with the ϵ-constraint method and only one reduction has to be made for each problem. Furthermore, we will see that by using a test set approach, the solution of redundant problems is avoided (that is, problems that produce efficient points already obtained) and only efficient solutions are obtained, instead of weakly efficient points that would have to be filtered in a second phase.

3. Main results

In order to treat the problem (1) with the ϵ-constraint method, we address problems of the form

$$\begin{array}{ll} \min & c_2(\mathbf{x}) \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} \\ & c_1(\mathbf{x}) \leq \epsilon_1, \quad \mathbf{x} \in \mathbb{Z}^n_{\geq 0}, \end{array} \tag{4}$$

or in standard form,

$$\begin{array}{ll} \min & c_2(\mathbf{x}) \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} \\ & c_1(\mathbf{x}) + s = \epsilon_1, \quad \mathbf{x} \in \mathbb{Z}^n_{\geq 0}, \end{array} \tag{5}$$

which will be denoted $P_2(\epsilon_1)$ (similarly, if the cost function is c_1 and the constraint is related to c_2, then it is denoted $P_1(\epsilon_2)$). A problem $P_2(\epsilon_1)$ can have several optimal solutions. Our purpose consists of obtaining among them a minimal complete set of efficient points as described in Ehrgott (2005), as outlined in Section 2.

Notation 1. We denote as $\mathcal{T}_{21} \subset \mathbb{Z}^{n+1}$ the test sets associated with the family of problems $P_2(\epsilon_1)$ for any RHS, with the properties pointed out in Theorem (2.2). The elements of the test set include one additional variable since a slack variable has been added to the problem to put it in standard form. Test sets will be assumed to be computed using Gröbner bases with respect to a total ordering $<_{c_2,c_1,\prec}$. This total ordering $<_{c_2,c_1,\prec}$ first compares two points using c_2, breaks ties with c_1 and with a total ordering $\prec$ if the points have the same costs c_1 and c_2. For example for $n = 3$, and c_1 and c_2 with coefficients $(1,1,1)$ and $(-1,1,-1)$, the points $(1,2,0)$ and $(0,2,1)$ have the same costs. If we choose the ordering $\prec = <_{\text{lex}}$ to break the ties, we obtain $(0,2,1) <_{c_2,c_1,<_{\text{lex}}} (1,2,0)$.

We use test sets $\mathcal{T}_{12}$ by simply changing the roles of c_1 and c_2. Solving $P_2(\epsilon)_{\mathcal{T}_{21}}$ involves solving $P_2(\epsilon)$ by specifically using $\mathcal{T}_{21}$, that is, the unique best solution of the problem $P_2(\epsilon)$ with respect to a total ordering of type $<_{c_2,c_1,\prec}$ is obtained. The selection of this particular ordering is of major interest for Theorem 3.1. This selection of the ordering provides a direct computation of the efficient points, in a single step: no second phase is needed in order to eliminate weakly efficient points.

The following result provides a characterization of the efficient points in this context.

Theorem 3.1. $(\mathbf{x}^\star,\ 0)$ *is the optimal solution of the problem* $P_2(c_1(\mathbf{x}^*))_{\mathcal{T}_{21}}$ *if and only if* $\mathbf{x}^\star$ *is an efficient solution of the bi-objective problem* (1) *and, among the efficient solutions with costs* $(c_1(\mathbf{x}^\star),\ c_2(\mathbf{x}^\star))$, *it is the smallest with respect to the ordering* $<_{c_2,c_1,\prec}$.

Proof. Let $(\mathbf{x}^\star,\ 0)$ be the optimal solution of $P_2(c_1(\mathbf{x}^\star))_{\mathcal{T}_{21}}$. In particular $(\mathbf{x}^\star,\ 0)$ is a feasible solution of $P_2(c_1(\mathbf{x}^\star))$. To show that $\mathbf{x}^\star$ is an efficient solution of the bi-objective problem using Theorem 2.1, it suffices to prove that $(\mathbf{x}^\star,\ 0)$ is an optimal solution of the problem $P_1(c_2(\mathbf{x}^\star))$, that is:

$$\begin{array}{ll} \min & c_1(\mathbf{x}) \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} \\ & c_2(\mathbf{x}) + s = c_2(\mathbf{x}^\star), \quad \mathbf{x} \in \mathbb{Z}^n_{\geq 0} \end{array} \tag{6}$$

We have that $(\mathbf{x}^\star,\ 0)$ is feasible for (6). Proving by contradiction, let us assume that $(\mathbf{x}^\star,\ 0)$ is not an optimal solution of this problem. Let $(\mathbf{x}',\ s)$ with $s \geq 0$ be an optimal solution of (6), hence $c_1(\mathbf{x}') < c_1(\mathbf{x}^\star)$ and $c_2(\mathbf{x}') + s = c_2(\mathbf{x}^\star)$, that is, $c_2(\mathbf{x}') \leq c_2(\mathbf{x}^\star)$. Thus, we have found that $(\mathbf{x}',\ s')$ for $s' = c_1(\mathbf{x}^\star) - c_1(\mathbf{x}')$ is feasible for $P_2(c_1(\mathbf{x}^\star))$ with $(\mathbf{x}', s') <_{c_2,c_1,\prec} (\mathbf{x}^\star, 0)$, which contradicts the fact of $(\mathbf{x}^\star,\ 0)$ being the optimal solution of $P_2(c_1(\mathbf{x}^\star))_{\mathcal{T}_{21}}$. Therefore, $(\mathbf{x}^\star,\ 0)$ is an optimal solution of (6) and, since the efficient solutions with the same costs as $\mathbf{x}^\star$ are feasible for $P_2(c_1(\mathbf{x}^\star))$, we deduce that $(\mathbf{x}^\star,\ 0)$ is the best of these solutions with respect to $<_{c_2,c_1,\prec}$.

Reciprocally, let $\mathbf{x}^\star$ be the efficient solution of (1) with costs $(c_1(\mathbf{x}^\star),\ c_2(\mathbf{x}^\star))$, and the smallest one with respect to $<_{c_2,c_1,\prec}$. From Theorem 2.1, we know that $\mathbf{x}^\star$ is feasible for $P_2(c_1(\mathbf{x}^\star))$. If there were several efficient solutions with the same costs, then we would choose $\mathbf{x}^\star$ to be the optimum with respect to $<_{c_2,c_1,\prec}$, that is, the solution of $P_2(c_1(\mathbf{x}^\star))_{\mathcal{T}_{21}}$. □

Theorem 3.1 specifically provides a way to obtain the first point of our set of representatives of the non-dominated set of points of problem (1): that with minimum c_1.

Corollary 3.1.1. *(Ehrgott, 2005, Lemma 5.2.) If* $\mathbf{x}^\star_1$ *is the optimal solution of*

$$\min\{c_1(\mathbf{x}) : A\mathbf{x} = \mathbf{b}, \mathbf{x} \in \mathbb{Z}^n_{\geq 0}\}$$

with respect to the ordering $<_{c_1,c_2,\prec}$, *then* $\mathbf{x}^\star_1$ *is an efficient solution of* (1) *with minimum* c_1.

Given the set of efficient solutions $\mathcal{X}$, let us denote $\mathcal{X}^\star \subset \mathcal{X}$ as the minimal complete set of efficient points whose elements have the property of being the smallest solutions with respect to $<_{c_2,c_1,\prec}$ from among the points that have the same costs, that is

$$\begin{aligned} \mathcal{X}^\star = \{\mathbf{x}^\star \in \mathcal{X} \,|\, &(\mathbf{x}^\star, 0) \text{ is a solution of} \\ &P_2(z_1)_{\mathcal{T}_{21}} \text{ for certain } (z_1, z_2) \in \mathcal{N}\} \end{aligned} \tag{7}$$

We order $\mathcal{X}^\star = \{\mathbf{x}^\star_1, \ldots, \mathbf{x}^\star_{|\mathcal{N}|}\}$ in a strictly increasing way with respect to c_1 (and hence in a strictly decreasing way in c_2) where $\mathbf{x}^\star_1$ is the result of Corollary 3.1.1. It is clear that each $(\mathbf{x}^\star_i, 0)$ is the solution of $P_2(c_1(\mathbf{x}^\star_i))_{\mathcal{T}_{21}}$ for $i = 1, \ldots, |\mathcal{N}|$ from Theorem 3.1. We now see how to proceed to obtain the elements of $\mathcal{X}^\star$ while avoiding calculations that would lead to efficient solutions previously computed. To this end, we denote for each $\mathbf{x} \in \mathbb{Z}^n_{\geq 0}$ the set

$$G_{\mathbf{x}} = \{(\mathbf{t}, s) \in \mathcal{T}_{21} : \mathbf{t} \leq \mathbf{x}\}.$$

It should be borne in mind that if $(\mathbf{x}, 0)$ is the optimal solution of $P_2(\epsilon)_{\mathcal{T}_{21}}$ for certain ϵ with $\mathbf{x} \in \mathbb{Z}^n_{\geq 0}$ and $G_{\mathbf{x}} \neq \emptyset$, then for every $(\mathbf{t}, s) \in G_{\mathbf{x}}$ we have $s > 0$, otherwise $(\mathbf{x}, 0)$ would be reducible.

Proposition 3.1. *For any $\mathbf{x}^\star_i$, $i = 1, \ldots, |\mathcal{N}| - 1$ we have that:*

1) *Let s_i be the smallest $s > 0$ such that $(\mathbf{t}, s) \in G_{\mathbf{x}^\star_i}$. For any s', $0 < s' < s_i$, the solution of $P_2(c_1(\mathbf{x}^\star_i) + s')_{\mathcal{T}_{21}}$ is $(\mathbf{x}^\star_i, s')$ and for that reason $P_2(c_1(\mathbf{x}^\star_i) + s')_{\mathcal{T}_{21}}$ does not provide a new efficient solution.*
2) *For s_i, which is the smallest $s > 0$ such that $(\mathbf{t}, s) \in G_{\mathbf{x}^\star_i}$, the solution of $P_2(c_1(\mathbf{x}^\star_i) + s_i)_{\mathcal{T}_{21}}$ is $(\mathbf{x}^\star_{i+1}, 0)$.*
3) *$G_{\mathbf{x}^\star_{|\mathcal{N}|}} = \emptyset$.*

Proof.

1) To begin with, we have $G_{\mathbf{x}^\star_i} \neq \emptyset$. We have that $\mathbf{x}^\star_i$, is not the optimal solution of $P_2(c_1(\mathbf{x}^\star_{i+1}))_{\mathcal{T}_{21}}$, but is feasible. There must be a value $s > 0$ such that $(\mathbf{x}^\star_i, s)$ is reducible to obtain better values of c_2 by subtracting elements of $\mathcal{T}_{21}$. If $s_i > 0$ is the smallest of such possible $s > 0$ and $0 < s' < s$, then the point $(\mathbf{x}^\star_i, s')$ is feasible for $P_2(c_1(\mathbf{x}^\star_i) + s')_{\mathcal{T}_{21}}$. If it is not the optimal solution, then there exists $(\mathbf{t}, s) \in \mathcal{T}_{21}$ such that $\mathbf{t} \leq \mathbf{x}^\star_i$ and $s \leq s' < s_i$, but this inequality contradicts the fact that s_i is the smallest one with this property.
2) Let us first see whether the solution of $P_2(c_1(\mathbf{x}^\star_i) + s_i)_{\mathcal{T}_{21}}$ is of the form $(\mathbf{x}', 0)$. The point $(\mathbf{x}^\star_i, s_i)$ is feasible for $P_2(c_1(\mathbf{x}^\star_i) + s_i)_{\mathcal{T}_{21}}$ but it is not the optimal solution, because there exists an element $(\mathbf{t}, s_i)$ in the test set such that $(\mathbf{t}, s_i) \leq (\mathbf{x}^\star, s_i)$ (componentwise), hence $(\mathbf{x}^\star_i - \mathbf{t}, 0)$ is feasible for $P_2(c_1(\mathbf{x}^\star_i) + s_i)_{\mathcal{T}_{21}}$ with $(\mathbf{x}^\star_i - \mathbf{t}, 0) <_{c_2, c_1, \prec} (\mathbf{x}^\star_i, s_i)$ and consequently $c_2(\mathbf{x}^\star_i - \mathbf{t}) \leq c_2(\mathbf{x}^\star_i)$. As $c_1(\mathbf{x}^\star_i - \mathbf{t}) = c_1(\mathbf{x}^\star_i) + s_i$ and $s_i > 0$ then $c_1(\mathbf{x}^\star_i) < c_1(\mathbf{x}^\star_i - \mathbf{t})$. Thus, by the definition of the ordering $<_{c_2, c_1, \prec}$ we conclude that $c_2(\mathbf{x}^\star_i - \mathbf{t}) < c_2(\mathbf{x}^\star_i)$. Now let us suppose that the solution of $P_2(c_1(\mathbf{x}^\star_i) + s_i)_{\mathcal{T}_{21}}$ is the point $(\mathbf{x}', s')$. Let us show that $s' = 0$. If $s' > 0$ then $c_1(\mathbf{x}') + s' = c_1(\mathbf{x}^\star_i) + s_i = c_1(\mathbf{x}^\star_i - \mathbf{t})$ and $c_2(\mathbf{x}') \leq c_2(\mathbf{x}^\star_i - \mathbf{t}) < c_2(\mathbf{x}^\star_i)$. For the comparison of $s' > 0$ and $s_i > 0$ there are two options. If $0 < s' < s_i$ using part 1) above, therefore the solution of $P_2(c_1(\mathbf{x}^\star_i) + s_i - s')_{\mathcal{T}_{21}}$ is $(\mathbf{x}^\star_i, s_i - s')$. As $(\mathbf{x}', 0)$ is feasible, it holds that $(\mathbf{x}^\star_i, s_i - s') <_{c_2, c_1, \prec} (\mathbf{x}', 0)$, then $c_2(\mathbf{x}^\star_i) \leq c_2(\mathbf{x}')$, but this is a contradiction. If $0 < s_i < s'$ then $(\mathbf{x}', s' - s_i)$ is feasible for $P_2(c_1(\mathbf{x}^\star_i))_{\mathcal{T}_{21}}$ because $c_1(\mathbf{x}') + s' - s_i = c_1(\mathbf{x}^\star_i)$. However, as $(\mathbf{x}^\star_i, 0)$ is the solution of this problem, we deduce that $(\mathbf{x}^\star_i, 0) <_{c_2, c_1, \prec} (\mathbf{x}', s' - s_i)$, hence $c_2(\mathbf{x}^\star_i) \leq c_2(\mathbf{x}')$, which is also a contradiction. Consequently, we have obtained that $s_i = s' = 0$ and the solution $P_2(c_1(\mathbf{x}^\star_i) + s_i)_{\mathcal{T}_{21}}$ is a point of the form $(\mathbf{x}', 0)$. Nevertheless, $\mathbf{x}' = \mathbf{x}^\star_{i+1}$ must hold because $(\mathbf{x}', 0)$ is the optimal solution of $P_2(c_1(\mathbf{x}')_{\mathcal{T}_{21}})$ and is therefore an element of $\mathcal{X}^\star$, precisely the solution with the next value of c_2.
3) When we obtain the efficient solution $\mathbf{x}^\star_{|\mathcal{N}|}$ (that with the greatest c_1 among the non-dominated points), it must be $G_{\mathbf{x}^\star_{|\mathcal{N}|}} = \emptyset$ because otherwise from part 2) we would produce a new efficient point with a greater c_1 than $\mathbf{x}^\star_{|\mathcal{N}|}$.

□

The following theorem specifies the values of the s_i of the above proposition.

Theorem 3.2. *Let $\mathcal{X}^\star = \{\mathbf{x}^\star_1, \ldots, \mathbf{x}^\star_{|\mathcal{N}|}\}$ be defined as in (7) with $c_1(\mathbf{x}^\star_i) < c_1(\mathbf{x}^\star_{i+1})$ for every $i = 1, \ldots, |\mathcal{N}| - 1$, and $\mathcal{T}_{21}$ be the test set associated with the family $P_2(\epsilon)$ with respect to the ordering $<_{c_2, c_1, \prec}$. We have that for every $i = 1, \ldots, |\mathcal{N}| - 1$ there exists an element $(\mathbf{t}_i, s_i) \in \mathcal{T}_{21}$ such that $\mathbf{t}_i \leq \mathbf{x}^\star_i$ (componentwise) and $s_i = c_1(\mathbf{x}^\star_{i+1}) - c_1(\mathbf{x}^\star_i)$.*

Proof. Given $\mathbf{x}^\star_r \in \mathcal{X}^\star$, from Theorem 3.1 $(\mathbf{x}^\star_i, 0)$ is the optimal solution of $P_2(c_1(\mathbf{x}^\star_i))_{\mathcal{T}_{21}}$, and hence it is not reducible, that is, there is no $(\mathbf{t}, s) \in \mathcal{T}_{21}$ with $s \geq 0$ such that $\mathbf{t} \leq \mathbf{x}^\star_i$. In the same way, $(\mathbf{x}^\star_{i+1}, 0)$ is the solution of $P_2(c_1(\mathbf{x}^\star_{i+1}))_{\mathcal{T}_{21}}$, and $(\mathbf{x}^\star_i, s_i)$ is feasible for this problem with $s_i = c_1(\mathbf{x}^\star_{i+1}) - c_1(\mathbf{x}^\star_i)$. Since $(\mathbf{x}^\star_i, s_i)$ is not the optimum for $P_2(c_1(\mathbf{x}^\star_{i+1}))_{\mathcal{T}_{21}}$, there must exist an element $(\mathbf{t}', s')$ in the test set whereby $(\mathbf{t}', s') \leq (\mathbf{x}^\star_i, s_i)$. Then $s' > 0$ must hold, otherwise $(\mathbf{t}', s')$ would reduce $(\mathbf{x}^\star_i, 0)$ which is irreducible.

Let us show that $s_i = s'$. In order to carry out a proof by contradiction, let us suppose that $s_i \neq s'$. Since $(\mathbf{t}', s') \leq (\mathbf{x}^\star_i, s_i)$, then $0 < s' < s_i$. If we consider the problem $P_2(c_1(\mathbf{x}^\star_i) + s')_{\mathcal{T}_{21}}$, then $(\mathbf{x}^\star_i, s')$ is feasible for this problem but not the solution because it is reducible by means of $(\mathbf{t}', s')$. Consequently, the point $(\mathbf{x}^\star_i - \mathbf{t}', 0)$ is feasible for $P_2(c_1(\mathbf{x}^\star_i) + s')_{\mathcal{T}_{21}}$ with $(\mathbf{x}^\star_i - \mathbf{t}_i, 0) <_{c_2, c_1, \prec} (\mathbf{x}^\star_i, s')$, which implies that $c_2(\mathbf{x}_i - \mathbf{t}') \leq c_2(\mathbf{x}^\star_i)$. Moreover, we can affirm that $c_1(\mathbf{x}_i) < c_1(\mathbf{x}^\star_i - \mathbf{t}') < c_1(\mathbf{x}^\star_{i+1})$ due to $c_1(\mathbf{x}^\star_i - \mathbf{t}') = c_1(\mathbf{x}^\star{}_i) + s'$ with $s' > 0$, and assuming that $s' < s_i$. This last inequality and the definition of $\mathcal{X}^\star$ lead us to confirm that there is no point in the Pareto frontier whose value of c_1 is equal to that of $\mathbf{x}^\star_i - \mathbf{t}'$. Accordingly, $\mathbf{x}^\star_i - \mathbf{t}'$ is dominated by an efficient solution $\mathbf{x}^\star_k$ of $\mathcal{X}^\star$ with $k \leq i$, since $c_1(\mathbf{x}^\star_i) < c_1(\mathbf{x}^\star_i - \mathbf{t}')$. Hence, $c_1(\mathbf{x}^\star_k) < c_1(\mathbf{x}^\star_i - \mathbf{t}')$ and $c_2(\mathbf{x}^\star_k) \leq c_2(\mathbf{x}^\star_i - \mathbf{t}')$, and therefore $c_2(\mathbf{x}^\star_i - \mathbf{t}') = c_2(\mathbf{x}^\star_i)$. Since $(\mathbf{x}^\star_i - \mathbf{t}', 0) <_{c_2, c_1, \prec} (\mathbf{x}_i, s')$, we conclude that $c_1(\mathbf{x}^\star_i - \mathbf{t}_i) \leq c_1(\mathbf{x}^\star_i)$, which leads us to a contradiction. Therefore $s_i = s'$. □

Remark 1. In general, the point of the form $(\mathbf{x}^\star_i, s_i) - (\mathbf{t}, s_i)$ of the last theorem is *not* the next efficient point: this point must be reduced by the test set. Consider the problem

$$\begin{array}{ll} \max & x_1, x_2 \\ \text{s.t.} & 3x_1 + 2x_2 \leq 21 \\ & x_1, x_2 \in \mathbb{Z}_{\geq 0}, \end{array} \tag{8}$$

or in standard form

$$\begin{array}{ll} \min & c_1 = -x_1, c_2 = -x_2 \\ \text{s.t.} & 3x_1 + 2x_2 + x_3 = 21 \\ & x_1, x_2, x_3 \in \mathbb{Z}_{\geq 0}. \end{array} \tag{9}$$

The optimal solution with respect to $<_{c_1, c_2, \prec}$ of

$$\begin{array}{ll} \min & -x_1 \\ \text{s.t.} & 3x_1 + 2x_2 + x_3 = 21 \\ & x_1, x_2, x_3 \in \mathbb{Z}_{\geq 0} \end{array} \tag{10}$$

is $\mathbf{x}^\star_1 = (7, 0, 0)$. We subsequently have to manage problems (already in standard form) of type

$$\begin{array}{ll} \min & -x_2 \\ \text{s.t.} & 3x_1 + 2x_2 + x_3 = 21 \\ & -x_1 + s = \epsilon_1 \\ & x_1, x_2, x_3, s \in \mathbb{Z}_{\geq 0}. \end{array} \tag{11}$$

The test set $\mathcal{T}_{21}$ for this problem (computed using (4ti2 team, 2015)) with respect to $<_{c_2, c_1, <_{\text{lex}}}$ is

$$\{(0, -1, 2, 0), (1, -1, -1, 1)\}.$$

We start from $(\mathbf{x}^\star_1, 0) = (7, 0, 0, 0)$, for which the corresponding s_1 (postulated in Proposition 3.1) is $s_1 = 1$. To solve $P_2(7)_{\mathcal{T}_{21}}$ we start from (7,0,0,1) and only the vector $(1, -1, -1, 1)$ has to be subtracted to obtain $(\mathbf{x}^\star_2, 0) = (6, 1, 1, 0)$, which is irreducible, and thus $\mathbf{x}^\star_2$ is a new efficient solution. From this non-dominated point we obtain $s_2 = 1$ and we need to solve $P_2(6)_{\mathcal{T}_{21}}$ by reducing (6,1,1,1), in this case

$$(6, 1, 1, 1) - (1, -1, -1, 1) = (5, 2, 2, 0)$$

The point (5,2,2,0) is a weakly efficient solution for this problem with costs $(-5, -2)$. This is not the smallest efficient solution with

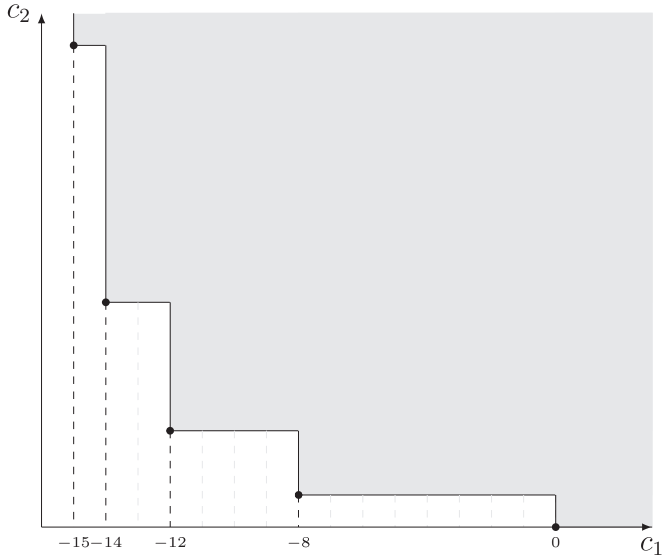

Fig. 1. Algorithm 1 on BBV for $n = 4$ provides the values $\epsilon = -15, -14, -12, -8, 0$ for which $P_2(\epsilon)$ provides the complete Pareto frontier, and avoids unnecessary values.

respect to $<_{c_2,c_1,<_{\text{lex}}}$. To obtain this representative element another reduction must be performed to obtain an efficient solution:

$$(5,2,2,0) - (0,-1,2,0) = (5,3,0,0) = (\mathbf{x}_3^\star, 0).$$

In order to complete this section the algorithm to solve problem (1) is presented. It is based on the classical ϵ-constraint method with the additional tool of algebraic test sets. Our algorithm computes a minimal set of efficient solutions and, for this purpose, it solves exactly one single-objective IP for each representative element, not by passing the IP to a solver (such as CPLEX), but by reducing with test sets. The correctness of the algorithm is proved in Corollary 3.1.1 and Proposition 3.1.

4. A working example

In Laumanns et al. (2006), the problem BBV (*Bi-objective Binary Value*) is presented. It is a bi-objective generalisation of the Binary Value problem proposed for the complexity analysis of evolutionary algorithms. It exemplifies a bi-objective problem depending on n variables, with a Pareto set of size $n+1$, but with an exponential number of problems to be solved if the ϵ-constraint technique is used.

The pseudo-Boolean function BBV is defined as

$$\text{BBV}(x_1, \ldots, x_n) = \left(\sum_{j=1}^{n} 2^{n-j} x_j, \sum_{j=1}^{n} 2^{j-1}(1 - x_j) \right)$$

Although the complexity of computing Gröbner basis is exponential in the number of variables, in this case we can give a closed formula for both test sets needed in our Algorithm 1. The proofs of these results are technical and completely described in Appendix A.

Our algorithm is illustrated by obtaining the Pareto frontier for the BBV family of problems for the case $n = 4$, which, in its standard form is

$$\begin{array}{ll} \min & -8x_1 - 4x_2 - 2x_3 - x_4, x_1 + 2x_2 + 4x_3 + 8x_4 \\ \text{s.t.} & x_j + r_j = 1 \ (j = 1, \ldots, 4) \end{array} \tag{12}$$

The feasible points are written with the x_j variables in the first places, and the slack variables r_j in the last ones. The test

Algorithm 1 Algorithm to obtain the exact Pareto frontier of a Bi-objective Linear Integer Problem.

```
INPUT: cost vector $c = (c_1, c_2)$, constraint matrix $A$ and vector $\mathbf{b}$ of the problem (1).
Compute the test set $\mathcal{T}_{12}$ associated with the family $P_1(\epsilon)$with respect to the ordering $<_{c_1,c_2,\prec}$.
Reduce a feasible point of $A\mathbf{x} = \mathbf{b}$ with respect to $\mathcal{T}_{12}$ to solve $\min\{c_1(\mathbf{x}) : A\mathbf{x} = \mathbf{b}, \mathbf{x} \in \mathbb{Z}^n_{\geq 0}\}$. Denote $\mathbf{x}_1^\star$ as the optimal solution and $c_1(\mathbf{x}_1^\star) = c_1^*$
$\mathcal{X}^\star := \{\mathbf{x}_1^\star\}$.
$\epsilon := c_1^*$.
Compute $\mathcal{T}_{21}$ the test set associated with the family $P_2(\epsilon)$ with respect to an ordering$<_{c_2,c_1,\prec}$
$G_{\mathbf{x}_1^\star} := \{(\mathbf{t}, s) \in \mathcal{T}_{21}$ such that $\mathbf{t} \leq \mathbf{x}_1^\star\}$.
$i := 1$
While $G_{\mathbf{x}_i^\star} \neq \emptyset$ Do:
    $s_i$ := the smallest $s > 0$ such that $(\mathbf{t}, s) \in G_{\mathbf{x}_i^\star}$.
    $\epsilon := \epsilon + s_i$.
    Reduce initial feasible point $(\mathbf{x}_i^\star, s_i)$ with $\mathcal{T}_{21}$ to solve $P_2(\epsilon)$. Denote $(\mathbf{x}_{i+1}^\star, 0)$ as the solution.
    Compute $G_{\mathbf{x}_{i+1}^\star}$.
    $\mathcal{X}^\star := \mathcal{X}^\star \cup \{\mathbf{x}_{i+1}^\star\}$.
    $i := i + 1$
End While
Output: $\mathcal{X}^\star$ such that $c(\mathcal{X}^\star) = \mathcal{N}$ as in (7).
```

set of the problem with respect to the ordering $<_{c_1,c_2,<_{\text{lex}}}$ where $c_1 = (-8, -4, -2, -1, 0, 0, 0, 0)$ and $c_2 = (1, 2, 4, 8, 0, 0, 0, 0)$ is

$$\mathcal{T}_{12} = \{(-1, 0, 0, 0, 1, 0, 0, 0), (0, -1, 0, 0, 0, 1, 0, 0),$$

$$(0, 0, -1, 0, 0, 0, 1, 0), (0, 0, 0, -1, 0, 0, 0, 1)\}$$

By starting from the feasible point (0,0,0,0,1,1,1,1) and reducing (subtracting) with the test set,the first efficient point $\mathbf{x}_1^\star = (1, 1, 1, 1, 0, 0, 0, 0)$ with costs $(-15, 15)$ is obtained.

Now we consider the family of problems of the ϵ-constraint method with the new slack variable s in the last place:

$$\begin{array}{ll} \min & x_1 + 2x_2 + 4x_3 + 8x_4 \\ \text{s.t.} & x_j + r_j = 1 \ (j = 1, \ldots, 4) \\ & -8x_1 - 4x_2 - 2x_3 - x_4 + s = \epsilon \end{array} \tag{13}$$

The test set associated with the family (13) with respect to $<_{c_2,c_1,<_{\text{lex}}}$, that is, that which solves $P_2(\epsilon)_{\mathcal{T}_{21}}$ is

$$\begin{aligned} \mathcal{T}_{21} = \{ & (-1, 0, 0, 1, 1, 0, 0, -1, -7), (-1, 0, 1, 0, 1, 0, -1, 0, -6), \\ & (-1, 1, 0, 0, 1, -1, 0, 0, -4), (0, -1, 0, 1, 0, 1, 0, -1, -3), \\ & (0, -1, 1, 0, 0, 1, -1, 0, -2), (0, 0, -1, 1, 0, 0, 1, -1, -1), \\ & (0, 0, 0, 1, 0, 0, 0, -1, \mathbf{1}), (0, 0, 1, 0, 0, 0, -1, 0, \mathbf{2}), \\ & (0, 1, 0, 0, 0, -1, 0, 0, \mathbf{4}), (1, 0, 0, 0, -1, 0, 0, 0, \mathbf{8})\}, \end{aligned}$$

where the boldface components indicate positive values for s. We have to start with $\epsilon = c_1(\mathbf{x}_1^\star) = c_1^\star = -15$, and our starting point is now $(\mathbf{x}_1^\star, 0) = (1, 1, 1, 1, 0, 0, 0, 0, 0)$. It can be observed that

$$\begin{aligned} G_{\mathbf{x}_1^\star} = \{ & (0, 0, 0, 1, 0, 0, 0, -1, \mathbf{1}), (0, 0, 1, 0, 0, 0, -1, 0, \mathbf{2}), \\ & (0, 1, 0, 0, 0, -1, 0, 0, \mathbf{4}), (1, 0, 0, 0, -1, 0, 0, 0, \mathbf{8})\} \end{aligned}$$

and then s_1 is the smallest $s > 0$ such that $(\mathbf{t}, s) \in G_{\mathbf{x}_1}$, that is, the element of the test set with the smallest s that reduces $(\mathbf{x}_1, s)$. In this case, this is $(0, 0, 0, 1, 0, 0, 0, -1, \mathbf{1})$, and hence $s_1 = 1$ and $\epsilon = -15 + 1 = -14$. We then solve $P_2(-14)_{\mathcal{T}_{21}}$ reducing (1, 1, 1, 1, 0, 0, 0, 0, 1) with $\mathcal{T}_{21}$, and $(\mathbf{x}_2^\star, 0) = (1, 1, 1, 0, 0, 0, 0, 1, 0)$ is obtained. In this case,

$$\begin{aligned} G_{\mathbf{x}_2^\star} = \{ & (0, 0, 1, 0, 0, 0, -1, 0, \mathbf{2}), (0, 1, 0, 0, 0, -1, 0, 0, \mathbf{4}), \\ & (1, 0, 0, 0, -1, 0, 0, 0, \mathbf{8})\} \end{aligned}$$

Table 1
Bi-objective unbounded knapsack problem for $n = 50$.

Type		$\lvert\mathcal{T}_{21}\rvert$	$\lvert\mathcal{N}\rvert$	Algorithm		AUGMECON2 (s)
				4ti2 (s)	t_r (s)	
A	average	104.03	140.43	0.0551	0.2684	10.9847
	max.	124	841	0.0608	1.4627	57.93
B	average	103.2	8.43	0.0551	0.0415	0.6773
	max.	137	185	0.0602	0.4513	16.3
C	average	107	326.53	0.0552	0.611	27.2177
	max.	149	1753	0.0573	3.2606	146.29
D	average	7280.63	1076.4	7.8185	134.4293	287.6793
	max.	12375	1896	17.0606	379.918	2132.22

and for that reason the vector $(0, 0, 1, 0, 0, 0, -1, 0, \mathbf{2})$ indicates that $s_2 = 2$ and $\epsilon = -14 + 2 = -12$. The subsequent problem to solve in order to find the next efficient point is $P_2(-12)_{\mathcal{T}_{21}}$ starting from (1,1,1,0,0,0,0,1,2).

In this way, we follow the algorithm to find the successive efficient solutions

$$\begin{aligned} & \mathbf{x}_3^\star = (1, 1, 0, 0, 0, 0, 1, 1), \\ & \mathbf{x}_4^\star = (1, 0, 0, 0, 0, 1, 1, 1) \\ \text{and} \quad & \mathbf{x}_5^\star = (0, 0, 0, 0, 1, 1, 1, 1). \end{aligned}$$

In the case of the last point, it is detected that point (0, 0, 0, 0, 1, 1, 1, 1, s) is irreducible for any s, and hence $G_{\mathbf{x}_5^\star}$ is empty and we have completed the calculation. The set of efficient points is (projected only on the initial variables of problem (12)):

$$\begin{aligned} \mathcal{X}^\star = \{ & (1, 1, 1, 1, 0, 0, 0, 0), (1, 1, 1, 0, 0, 0, 0, 1, 0), \\ & (1, 1, 0, 0, 0, 0, 1, 1, 0), (1, 0, 0, 0, 0, 1, 1, 1, 0), \\ & (0, 0, 0, 0, 1, 1, 1, 1, 0) \ \} \end{aligned}$$

Their respective costs produce the Pareto frontier

$$\mathcal{N} = \{(-15, 15), (-14, 7), (-12, 3), (-8, 1), (0, 0)\}.$$

5. Computational experiments

We compare the performance of our algorithm with the highly cited algorithm AUGMECON2 presented in Mavrotas and Florios (2013) which is based on the ϵ-constraint method.

Our Algorithm 1 has been implemented in C++ and calls 4ti2 (see 4ti2 team (2015)) to compute the test sets. Our experiments have been conducted in a 3.7 GHz Quad-Core Intel Xeon with 12 GB RAM under OSX version 10.9.5. The algorithm AUGMECON2, written in GAMS, has been executed in GAMS 28.4 and calls CPLEX 12.7 to solve the single-objective problems, except for the examples of Table 6. For these bigger examples we have runned AUGMECON2 using CPLEX in `neos-server.org`, (see Czyzyk et al., Dolan (2001), Gropp and Moré (1997)).

The most promising results for our algorithm are obtained in the *multi-objective unbounded knapsack problem*. As mentioned in Rong and Figueira (2013), this case is interesting since it is more difficult than the binary case, a priori. The problem for two objectives has the form

$$\begin{aligned} \max \quad & \sum_{j=1}^{n} c_{kj} x_j \quad k = 1, 2 \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_j x_j \le b, \ \ x_j \in \mathbb{Z}_{\ge 0} \end{aligned} \tag{14}$$

where b is the limit of weight for the knapsack and there are n available objects, each with weight a_j $(j = 1, \dots, n)$ and costs c_{kj}, one for each objective $k = 1, 2$. We assume that c_{kj}, a_j and b are positive integers.

We have managed instances of four types of problems, as considered in Bazgan, Hugot, and Vanderpooten (2009), which have been randomly generated with different options for the correlation between weights and costs: type A stands for random instances; B for positively correlated objective functions (unconflicting instances); C for negatively correlated objective functions (conflicting instances); and D for negatively correlated objective functions and correlated weight (conflicting instances with correlated weight). In all the examples, we have chosen $b = \lfloor \frac{1}{2} \sum_{j=1}^{n} a_j \rfloor$.

In all the following tables, after running 30 instances, we report:

1. The size of the Pareto frontier ($\lvert\mathcal{N}\rvert$).
2. The size of the second test set required for the problem ($\lvert\mathcal{T}_{21}\rvert$).
3. The CPU used by Algorithm 1 to obtain $\mathcal{N}$ is separated into the time taken by 4ti2 to generate the test sets (4ti2) and the remaining CPU time (t_r).
4. The CPU time used by AUGMECON2.

Several remarks should be made:

- In Table 1 the first comparison between our method and AUGMECON2 is presented for the bi-objective unbounded knapsack problem. The behaviour of AUGMECON2 is excellent in regards to the ratio number of single-objective problems/$\lvert\mathcal{N}\rvert$, which is very close to 1, therefore the use of algorithms as the ones presented in Zhang and Reimann (2014) or Kirlik and Sayın (2014) would not likely produce substantial improvements. Tables 1–5 show that reducing with test sets is definitely faster (between 30 and 50 times, except in case D) than solving single-objective problems with CPLEX with a assortment of branch and cut strategies, as long as the size of the test set is reasonable.
- In Table 1, we observe that the running time for Algorithm 1 is related to the size of $\mathcal{N}$ and the size of the test sets of the problem. As soon as the size of the test sets becomes huge, as it is the case of problems of type D with correlation, our method performs comparatively poorly. To reduce an element or to generate the sets $G_{\mathbf{x}}$, valid elements in the test set must be sought; this task obviously takes a greater effort in large test sets. At the same time, test sets of large size can be related to much more likely augmentations of values of ϵ for consideration, and hence the bigger the size of the test set, the longer the time to generate the set of efficient solutions.
- In Rong and Figueira (2014), the bi-objective unbounded knapsack problem is specifically treated with a dynamic programming approach that improves on previous work by the same authors (see Rong & Figueira (2013)). This approach achieves much better results than ours when n does not exceed 50, but when the size increases they recognize that their dynamic programming approach cannot handle the problems.

  In Table 2 we show the results of computing instances of the most difficult type D, for n up to 100. Our CPU times are quite

Table 2
Bi-objective unbounded knapsack problem of type D for $n = 60, \ldots, 100$.

n		$\|\mathcal{T}_{21}\|$	$\|\mathcal{N}\|$	Algorithm		AUGMECON2 (s)
				4ti2 (s)	t_r (s)	
50	average	7902.5	996.8	11.58	168.22	273.76
	max.	22976	2350	63.5	875.19	1576.78
60	average	8606.5	1427.4	14.48	245.6	269.39
	max.	21254	2498	62.34	921.38	619.38
70	average	10393.9	1636.8	22.18	438.54	254.02
	max.	22118	3280	81.03	1607.03	493.17
80	average	11944.8	1854.4	32.69	591.9	743.02
	max.	24130	3176	118.6	2033.61	2585.25
90	average	12517.5	2405.5	33.54	811.3	1518.42
	max.	14731	3237	53.54	1223.69	3748.86
100	average	21934.7	4228.2	11.227	2591.42	2846.89
	max.	29909	5244	266.08	4052.31	8724.22

Table 3
Bi-objective 0–1 Knapsack Problem (type A).

n		$\|\mathcal{T}_{21}\|$	$\|\mathcal{N}\|$	Algorithm		AUGMECON2 (s)
				4ti2 (s)	t_r (s)	
10	average	167.95	3.8	0.0823	0.0104	0.374
	max	271	10	0.1034	0.0261	0.9
15	average	1032.85	6.85	0.3535	0.0852	0.6705
	max	1681	12	1.6397	0.1543	1.16
20	average	3621.35	11.25	21.9757	0.5657	1.0755
	max	9664	17	399.541	1.8894	1.57
25	average	10556	15.95	12.639	2.4401	1.5355
	max	26987	27	66.7733	9.0831	2.5
30	average	27381.5	19.5	98.1166	7.9825	1.8635
	max	62379	35	390.58	20.7485	3.34

Table 4
Bi-objective unbounded knapsack problem for $m = 2$.

n, m		$\|\mathcal{T}_{21}\|$	$\|\mathcal{N}\|$	Algorithm		AUGMECON2 (s)
				4ti2 (s)	t_r (s)	
$n = 20,\ m = 2$	average	196.37	18.53	0.05	0.09	1.38
	max.	**1093**	**102**	**0.12**	**1.19**	**7.39**
$n = 25,\ m = 2$	average	446.17	31.97	0.11	0.31	2.49
	max.	**3401**	**129**	**1.05**	**5.06**	**11.54**
$n = 30,\ m = 2$	average	416.23	35.63	0.08	0.28	2.63
	max.	79	**206**	0.05	0.21	**13.86**
	max.	**2647**	92	**0.45**	**3.06**	7.45
$n = 35,\ m = 2$	average	441.10	45.81	0.08	0.48	3.39
	max.	1349	**244**	0.17	**4.91**	**18.79**
	max.	**1467**	66	**0.23**	1.44	5.15
$n = 40,\ m = 2$	average	527	65.97	0.11	0.72	4.81
	max.	523	**208**	0.09	1.66	**14.90**
	max.	**2622**	93	**0.56**	**3.80**	7.97

similar to those of AUGMECON2, in spite of the big sizes of some test sets that slows down our approach.

- If we consider the *binary* knapsack problem (see Table 3), that most widely treated in the literature, our method presents a clear disadvantage. The reason is that we have to add as many constraints of type $x_i + t_i = 1$ for each $x_i \in \{0,\ 1\}$ and the computation of the Gröbner bases that are behind our method is very sensitive to the number of restrictions and variables (cf. De Loera, Hemmecke, & Köppe (2013)). The running times are better with the use of the so-called *truncated Gröbner bases* (see Thomas & Weismantel (1997)), and it remains a work in progress to improve our running times with our method in the binary case.
- We have also studied the *multi-knapsack* problem. This is a generalisation of the classic problem in which several knapsacks are considered. It has the form

$$\begin{aligned} \max \quad & \sum_{j=1}^{n} c_{jk} x_j \quad k = 1, 2 \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_{lj} x_j \le b_l \quad l = 1, \ldots, m; \ \ x_j \in \mathbb{Z}_{\ge 0} \end{aligned} \tag{15}$$

In Tables 4 and 5, the results for 2 and 3 knapsacks are shown. test sets of these problems enlarge as the number of constraints (knapsacks) increases. Table 5 highlights how Algorithm 1 becomes worse than AUGMECON2 in certain cases. We have provided details of maximum values for the number of elements in the test set, number of non-dominated points, and running times. These maxima do not always match. We have used bold typeface to mark which of the parameters reach their maximum value.

Table 5
Bi-objective unbounded knapsack problem for $m = 3$.

n, m		$\|\mathcal{T}_{21}\|$	$\|\mathcal{N}\|$	Algorithm		AUGMECON2 (s)
				4ti2 (s)	t_r (s)	
$n = 20,\ m = 3$	average	3137.43	21.20	0.79	0.98	1.64
	max.	8872	**51**	3.51	**4.9**	**4.18**
	max.	**10177**	23	**4.51**	2.50	1.71
$n = 25,\ m = 3$	average	3349.33	25.47	1.41	1.29	1.88
	max.	2166	**131**	0.33	3.36	**9.60**
	max.	**15594**	34	**12.52**	6.53	2.58
	max.	14081	42	11.33	**7.09**	3.09
$n = 30,\ m = 3$	average	4468.70	32.60	2.11	2.38	2.52
	max.	5796	**91**	1.94	6.65	**7.34**
	max.	**17257**	5	**17.73**	1.95	0.44
	max.	12392	88	9.21	**13.75**	7.13
$n = 35,\ m = 3$	average	4674.2	38.35	2.62	4.32	3.01
	max.	11347	**125**	7.88	**21.60**	**9.36**
	max.	**14378**	55	**15.02**	12.32	5.00

Table 6
Larger examples of the multi-objective Unbounded Knapsack Problem (type A).

n		$\|\mathcal{N}\|$	Algorithm		AUGMECON2 (s)	
			4ti2 (s)	t_r (s)	grid $\approx \|\mathcal{N}\|$	grid $\approx$ range c_2
100	average	836.6	0.09	5.2	203.05	7940.2
	max	5745	0.10	36.77	1801.95	16215.9
200	average	5091.9	0.31	182.39	508.3	17144.6
	max	49358	0.68	2867.73	5783.4	‡
300	average	11110.23	0.86	715.04	3510.1	‡
	max	111172	2.41	9198.85	12809.2†	‡
400	average	21736.47	1.60	3073.73	6422.6†	‡
	max	97552	5.70	20760.1	16440.8†	‡
500	average	21245.97	4.78	3093.53	3145.9†	‡
	max	122182	13.96	14066.6	†	‡

- In Table 6, we present information on the limits of our method, while considering problems of $100, 200, \ldots, 500$ variables as benchmarks for future studies. Indeed, examples of up to 1000 variables have been made but, with the data considered, the majority produced a Pareto frontier of one single element.
  We have obtained remarkable results for these bigger examples in the comparison between our algoritm and AUGMECON2. Actually, we have runned two versions of AUGMECON2:
  - The first version, with a number of grid points around $|\mathcal{N}|$. This change in the algorithm produces experimentally much more faster running times and produces the exact set of non-dominated points for smaller examples (up to 200 variables, no more than 10,000 points in $\mathcal{N}$). Unfortunately, in many examples for 300, 400 or 500 variables the set of non-dominated points obtained is not complete (or even contains some weak Pareto points). The worst cases produce only about half of the points of the Pareto frontier. These cases correspond to the running times with † as a superindex.
  - The second version, the original algorithm as its is presented in Mavrotas and Florios (2013). The number of grid points is calculated as the range (difference between max and min) of the second objective function from the payoff table. As soon as we try big examples of 200 variables the examples do not finish in less than 8 hours that is the limit of time that Neos-server assigns to the submitted jobs. These cases are marked with a ‡ in the table.

6. Conclusions

We have introduced a new exact algorithm to obtain the set of non-dominated points of a bi-objective linear integer problem. It is based on the classic ϵ-constraint method but with an algebraic ingredient: the family of single-objective problems is considered using test sets. These test sets are computed via Gröbner bases with respect to an order, which, properly chosen, guides us in the process of obtaining only efficient solutions. The use of this test set approach has the following advantages compared to the classic ϵ-constraint method: 1) no unnecessary IPs are considered to obtain efficient solutions; 2) IPs are solved through the reduction of the points with only one test set for every problem, not calling solvers such as CPLEX; and 3) the reductions produce directly efficient solutions: no extra phase is required to discard weakly efficient solutions.

The computational experiments are promising, especially in the case of the unbounded knapsack problems which could be hard to treat with the techniques of the binary case. Our tables show that, in this family of unbounded knapsack problems, our algorithm performs better than such a sound and widely used algorithm as AUGMECON2, particularly, in examples with hundreds of variables. The performance is not that competitive in cases as the 0–1 knapsack and multiknapsack problems. It is apparent that the size of the test set is the main disadvantage to treat these cases with our approach.

It is a work in progress the generalisation of our test-set approach to multi-objective optimization for any number of objectives, in order to obtain the advantages of the bi-objective case as well.

Appendix A. On the calculation a priori of the test set for the family BBV

Although the computation of test sets via Gröbner bases can be potentially very expensive (or even impossible), the possibility of finding a closed formula for the test set *a priori* for a specific problem can be considered. It is in the case of the family of examples BBV that we have treated in this paper. We present in this appendix the results that are required to obtain such a formula for this particular family.

Recall that the BBV problem has the form

$$\begin{aligned}&\min \quad c_1 = \sum_{j=1}^{n} -2^{n-j}x_j,\, c_2 = \sum_{j=1}^{n} -2^{j-1}(1-x_j)\\ &\text{s.t.} \quad x_j \in \{0,1\}\end{aligned} \tag{A.1}$$

We denote r_j the slack variables for the constraints $x_j \leq 1$ in its standard form. Thus the constraint matrix of the problem is

$(I_n \; I_n)$

The following result characterizes the test set associated with the first step of the algorithm, that is, with the problem

$$\begin{aligned}&\min \quad c_1 = \sum_{j=1}^{n} -2^{n-j}x_j\\ &\text{s.t.} \quad x_j \in \{0,1\}\end{aligned} \tag{A.2}$$

with respect to $<_{c_1,c_2,\prec}$ for $\prec = <_{\text{lex}}$.

Proposition A.1. *Given the problem*

$$\begin{aligned}&\min \sum_{j=1}^{n} -2^{n-j}x_j\\ &\text{s.t. } x_j + r_j = 1,\end{aligned} \tag{A.3}$$

the associated test set with respect to the ordering $<_{c_1,c_2,\prec}$, induced by the costs

$$c_1 = (-2^{n-1}, -2^{n-2}, \ldots, -1, 0, \ldots, 0),$$
$$c_2 = (1, 2, 2^2, \ldots, 2^{n-1}, 0, \ldots, 0)$$

and a lexicographic ordering to break ties is

$$\mathcal{T}_{12} = \bigcup_{j=1}^{n} \{\underline{r_j} - x_j\}$$

with $j = 1, \ldots, n$

Proof. First we check that the elements of $\mathcal{T}_{12}$, belong to the kernel of $A = (I_n \; I_n)$ and that $x_j <_{c_1,c_2,\prec} r_j$. To prove that $\mathcal{T}_{12}$ is a test set associated with the reduced Gröbner of the toric ideal of matrix A, is enough to prove that given $\mathbf{z} = \mathbf{z}^+ - \mathbf{z}^- \in \mathbb{Z}^{2n}$ such that $\mathbf{z} \in Ker(A)$ where $\mathbf{z}^+$ stands for the leading term, that is, $\mathbf{z}^- <_{c_1,c_2,\prec} \mathbf{z}^+$, there exists an element in the test set $r_j - x_j$ such that $t_j \leq \mathbf{z}^+$ (cf. Natraj, Tayur, and Thomas (1995) for example).

For a vector $\mathbf{u} \in \mathbb{Z}^{2n}$ let us denote its first n components by $\mathbf{u}_{x_j}$ for $1 \leq j \leq n$ and its last n components by $\mathbf{u}_{r_j}$ for $1 \leq j \leq n$. If $\mathbf{u} \in Ker(A)$ then $\mathbf{u}_{x_i} = -\mathbf{u}_{r_j}$.

Let $\mathbf{z} = \mathbf{z}^+ - \mathbf{z}^-$ be an element in $Ker(A)$. If its last n components of $\mathbf{z}$ are equal to 0, that, $\mathbf{z}_{r_j} = 0$ for every $1 \leq j \leq n$, then $\mathbf{z} \in Ker(A_\mathbf{x})$ where $A_\mathbf{x} = (I_n \; 0)$, hence $\mathbf{z} = \mathbf{0}$.

So, let us suppose that some of the last n components of $\mathbf{z}$ are positive. Let p be the first index such that $\mathbf{z}_{r_p} > 0$. Then $\mathbf{z}^+_{r_p} > 0$ and $r_p \leq \mathbf{z}^+$.

Let us now suppose that all the last n components that are not null of $\mathbf{z}$ are less than or equal to zero and let p be the first index such that $\mathbf{z}_{r_p} < 0$. Then $\mathbf{z}^+$ has its last n components equal to 0, that is, $\mathbf{z}^+_{r_j} = 0$ for every $1 \leq j \leq n$ and $\mathbf{z}^-$ has its first n components equal to 0, that is to say, $\mathbf{z}^-_{x_j} = \mathbf{0}$ for every $1 \leq j \leq n$. Thus $c_1(\mathbf{z}^+) < 0$ and $c_1(\mathbf{z}^-) = 0$ but this leads us to a contradiction because $\mathbf{z}^+$ is the leading term. □

The next proposition provides the second test set for the family of problems BBV:

Proposition A.2. *Let us consider*

$$\begin{aligned}&\min \quad \sum_{j=1}^{n} -2^{j-1}(1-x_j)\\ &\text{s.t.} \quad x_j + r_j = 1\\ &\sum_{j=1}^{n} -2^{n-j}x_j + s = \epsilon\end{aligned} \tag{A.4}$$

A test set with respect to the ordering $<_{c_2,c_1,\prec}$ induced by the costs

$$c_2 = (1, 2, 2^2, \ldots, 2^{n-1}, 0, \ldots, 0),$$
$$c_1 = (-2^{n-1}, -2^{n-2}, \ldots, -1, 0, \ldots, 0)$$

and a lexicographic ordering to break ties (where r_j are the slack variables for the n first constraints and s is the slack variable for the last constraint) is:

$$\mathcal{T}_{21} = \bigcup_{j=1}^{n} \{\underline{x_j r_l} - x_l r_j s^{2^{n-l}-2^{n-j}}, \underline{x_j s^{2^{n-j}}} - r_j\}$$

with $1 \leq j$, $l \leq n$ and $l < j$,

Proof. It is clear that every element of $\mathcal{T}_{21}$ (leaders are underlined) belongs to the kernel of the matrix of constraints:

$$A = \begin{pmatrix} I_n & I_n & 0\\ c_1 & 0 & 1\end{pmatrix}$$

Given a vector $\mathbf{u} \in \mathbb{Z}^{2n+1}$ we denote its first n components as $\mathbf{u}_{x_j}$ with $1 \leq j \leq n$, the next n components as $\mathbf{u}_{r_j}$ with $1 \leq j \leq n$, and the last component as $\mathbf{u}_s$. If $\mathbf{u} \in Ker(A)$ then $\mathbf{u}_{x_j} = -\mathbf{u}_{r_j}$ and $\mathbf{u}_s = 2^{n-1}\mathbf{u}_{x_1} + 2^{n-2}\mathbf{u}_{x_2} + \cdots + \mathbf{u}_{x_n}$.

Let $\mathbf{z} = \mathbf{z}^+ - \mathbf{z}^-$ be an element of $\mathbb{Z}^{2n+1}$ such that $\mathbf{z} \in Ker(A)$ and $\mathbf{z}^+$ is the leading term with respect to $\preceq_{c_2,c_1}$, that is, $\mathbf{z}^- <_{c_2,c_1,\prec} \mathbf{z}^+$. We will show that it is reducible by an element of $\mathcal{T}_{21}$, that is to say, that the leader term of such an element is less or equal to $\mathbf{z}^+$.

If $\mathbf{z}^+_{x_j} = 0$, for every $1 \leq j \leq n$, that is, the vector $\mathbf{z}^+$ has its first n components equal to 0, then $c_2(\mathbf{z}^+) = 0$ but as we have that $c_2(\mathbf{z}^+) \geq c_2(\mathbf{z}^-)$, we obtain $\mathbf{z}^-_{x_j} = 0$ for every $1 \leq j \leq n$. Thus $\mathbf{z}_{x_j} = 0$ and, as $\mathbf{z}_{x_j} = -\mathbf{z}_{r_j}$ and $\mathbf{z}_s$ is a linear combination of the $\mathbf{z}_{x_j}$, we conclude that $\mathbf{z} = \mathbf{0}$.

If any of the first n components $\mathbf{z}^+$ is positive, let j' be the first index such that $\mathbf{z}^+_{x_{j'}} > 0$, so $\mathbf{z}^-_{r_{j'}} = \mathbf{z}^+_{x_{j'}} > 0$, given that $\mathbf{z} = \mathbf{z}^+ - \mathbf{z}^- \in Ker(A)$ and $\mathbf{z}^+$, $\mathbf{z}^-$ have disjoint support, it holds that $\mathbf{z}^+_{x_{j'}} = \mathbf{z}^-_{r_{j'}}$ and $-2^{n-1}\mathbf{z}^+_{x_1} - 2^{n-2}\mathbf{z}^+_{x_2} - \cdots - \mathbf{z}^+_{x_n} + \mathbf{z}_s = -2^{n-1}\mathbf{z}^-_{x_1} - 2^{n-2}\mathbf{z}^-_{x_2} - \cdots - \mathbf{z}^-_{x_n}$. As the last equality must hold, we have two possibilities: there exist another index l such that $\mathbf{z}^-_{x_l} > 0$ or $\mathbf{z}_s > 0$.

1. Let us suppose that there exist another index l such that $\mathbf{z}^-_{x_l} > 0$. If $l < j'$, then $\mathbf{z}^+_{r_l} = \mathbf{z}^-_{x_l} > 0$ and accordingly $x_{j'}r_l \leq \mathbf{z}^+$, so it is divided by an element of the Gröbner basis.
   If $j' < l$, let us note the components $l_1, \ldots, l_p$ in the variables x, in $\mathbf{z}^-$ and let us remark that $j < l_k$ for every $1 \leq k \leq p$, because if there exists $\mathbf{z}^+_{x_{j'}} > 0$ such that $l < j'$ then as $z^+_{r_l} > 0$, we would have $x_{j'}r_l \leq \mathbf{z}^+$. In this way, let $j_1, \ldots, j_r$ be such that $\mathbf{z}^+_{x_{j_n}} > 0$ and $l_1, \ldots, l_p$ such that $\mathbf{z}^-_{x_{l_k}} > 0$ with $j_1 < j_2 < \cdots < j_r < l_1 < l_2 < \cdots < l_p$. As $\mathbf{z} = \mathbf{z}^+ - \mathbf{z}^- \in Ker(A)$, the last row of A indicates that it has to be
   $$\begin{aligned}&-2^{n-i_1}\mathbf{z}^+_{x_{j_1}} - 2^{n-i_2}\,\mathbf{z}^+_{x_{j_2}} - \cdots - 2^{n-j_r}\mathbf{z}^+_{x_{j_r}} + 2^{n-l_1}\mathbf{z}^-_{x_{l_1}}\\ &+ 2^{n-l_2}\mathbf{z}^-_{x_{l_2}} + \cdots + 2^{n-l_p}\mathbf{z}^-_{x_{l_p}} + \mathbf{z}_s = 0\end{aligned}$$
   with $\mathbf{z}_s = 2^{n-j_1}\mathbf{z}^+_{x_{j_1}} + \cdots + 2^{n-j_r}\mathbf{z}^+_{x_{j_r}} - 2^{n-l_1}\mathbf{z}^-_{x_{l_1}} - \cdots - 2^{n-l_p}\mathbf{z}^-_{x_{l_p}}$.

If we prove that $\mathbf{z}_s \geq 2^{n-i_1}$ we will have $x_j c^{2^{n-j_1}} \leq \mathbf{z}^+$, but this is equivalent to prove that

$$2^{l_p-j_1}\mathbf{z}^+_{x_{j_1}} + \cdots + 2^{l_p-j_r}\mathbf{z}^+_{x_{jr}} - 2^{l_p-l_1}\mathbf{z}^-_{x_{l_1}} - \cdots - \mathbf{z}^-_{x_{l_p}} \geq 2^{l_p-j_1} \quad (A.5)$$

As $c_2(\mathbf{z}^+) \geq c_2(\mathbf{z}^-)$ we have that

$$\begin{aligned}&2^{j_1-1}\mathbf{z}^+_{x_{j_1}} + 2^{j_2-1}\mathbf{z}^+_{x_{j_2}} + \cdots + 2^{j_r-1}\mathbf{z}^+_{x_{jr}}\\ &\quad \geq 2^{l_1-1}\mathbf{z}^-_{x_{l_1}} + 2^{l_2-1}\mathbf{z}^-_{x_{l_2}} + \cdots + 2^{l_p-1}\mathbf{z}^-_{x_{l_p}}\end{aligned}$$

which implies

$$\max\{\mathbf{z}^+_{x_{j_1}}, \ldots, \mathbf{z}^+_{x_{jr}}\} \geq \sum_{k=1}^{p} \mathbf{z}^-_{x_{j_k}}$$

To do a proof by contradiction, let us suppose that $\max\{\mathbf{z}_{x_{j_1}}, \ldots, \mathbf{z}_{x_{jr}}\} < \sum_{k=1}^{p} \mathbf{z}_{x_{l_k}}$:

$$\begin{aligned}&2^{j_1-1}\mathbf{z}^+_{x_{j_1}} + 2^{j_2-1}\mathbf{z}^+_{x_{j_2}} + \cdots + 2^{j_r-1}\mathbf{z}^+_{x_{jr}}\\ &\quad \leq (2^{j_1-1} + \ldots + 2^{j_r-1})\max\{\mathbf{z}_{x_{j_1}}, \ldots, \mathbf{z}_{x_{jr}}\}\\ &\quad < (2^{j_1-1} + \ldots + 2^{j_r-1})\sum_{k=1}^{p}\mathbf{z}^-_{x_{l_k}} =\\ &\sum_{k=1}^{r}(2^{j_k-1})\mathbf{z}^-_{x_{l_1}} + \ldots + \sum_{k=1}^{r}(2^{j_k-1})\mathbf{z}^-_{x_{l_p}}\\ &\quad < 2^{l_1-1}\mathbf{z}^-_{x_{l_1}} + \ldots + 2^{l_p-1}\mathbf{z}^-_{x_{l_p}}.\end{aligned}$$

Using that $\sum_{k=1}^{t} 2^k < 2^{t+1}$ leads us to a contradiction. Let us note $\mathbf{z}^+_{x_{j_s}} = \max\{\mathbf{z}^+_{x_{j_1}}, \ldots, \mathbf{z}^+_{x_{jr}}\}$ and assume that $s \neq 1$. So

$$\begin{aligned}&2^{l_p-j_1}\mathbf{z}^+_{x_{j_1}} + \cdots + 2^{l_p-j_r}\mathbf{z}^+_{x_{jr}} - 2^{l_p-l_1}\mathbf{z}^-_{x_{l_1}} - \cdots - \mathbf{z}^-_{x_{l_p}}\\ &\quad \geq 2^{l_p-l_1}\mathbf{z}^+_{x_{j_1}} + \cdots + 2^{l_p-j_s}\sum_{k=1}^{p}\mathbf{z}^-_{x_{l_k}}\\ &\quad + \ldots + 2^{l_p-j_r}\mathbf{z}^+_{x_{jr}} - 2^{l_p-l_1}\mathbf{z}^-_{x_{l_1}} - \cdots - \mathbf{z}^-_{x_{l_p}}\\ &\quad = 2^{l_p-j_1}z^+_{x_{j_1}} + \underbrace{\cdots + (2^{l_p-j_s} - 2^{l_p-l_1})\mathbf{z}^-_{\mathbf{x}_{l_1}} + \ldots + (2^{l_p-j_s} - 1)\mathbf{z}^-_{x_{l_p}}}_{\geq 0}\\ &\quad \geq 2^{l_p-j_1}\end{aligned}$$

which finishes the proof of Eq. A.5 in this case.
If $r = 1$ as $c_2(\mathbf{z}^+) \geq c_2(\mathbf{z}^-)$ we have that

$$2^{j_1-1}\mathbf{z}^+_{x_{j_1}} \geq 2^{l_1-1}\mathbf{z}^-_{x_{l_1}} + 2^{l_2-1}\mathbf{z}^-_{x_{l_2}} + \cdots + 2^{l_p-1}\mathbf{z}^-_{x_{l_p}}$$

that is

$$\mathbf{z}^+_{j_1} \geq 2^{l_1-j_1}\mathbf{z}^-_{x_{l_1}} + 2^{l_2-j_1}\mathbf{z}^-_{x_{l_2}} + \cdots + 2^{l_p-j_1}\mathbf{z}^-_{x_{l_p}}$$

So

$$\begin{aligned}&2^{l_p-j_1}\mathbf{z}^+_{x_{j_1}} - 2^{l_p-l_1}\mathbf{z}^-_{x_{l_1}} - \cdots - \mathbf{z}^-_{x_{l_p}}\\ &\quad \geq 2^{l_p-j_1}(2^{l_1-j_1}\mathbf{z}^-_{x_{l_1}} + 2^{l_2-j_1}\mathbf{z}^-_{x_{l_2}} + \cdots + 2^{l_p-j_1}\mathbf{z}^-_{x_{l_p}})\\ &\qquad - 2^{l_p-l_1}\mathbf{z}^-_{x_{l_1}} - \cdots - \mathbf{z}^-_{x_{l_p}}\\ &\quad = (2^{l_p+l_1-2j_1} - 2^{l_p-j_1})\mathbf{z}^-_{x_{l_1}} + (2^{l_p+l_2-2j_1} - 2^{l_p-l_2})\mathbf{z}^-_{x_{l_2}}\\ &\qquad + \ldots + (2^{2(l_p-j_1)} - 1)\mathbf{z}^-_{x_{l_p}}\\ &\quad \geq (2^{2(l_p-j_1)} - 1)\mathbf{z}^-_{x_{l_p}} \geq 2^{2(l_p-j_1)} - 1 \geq 2^{l_p-j_1}\end{aligned}$$

If $r > 1$ and $x_{j_1} = \max\{\mathbf{z}^+_{x_{j_1}}, \ldots, \mathbf{z}^+_{x_{jr}}\}$:

$$\begin{aligned}&2^{l_p-j_1}\mathbf{z}^+_{x_{j_1}} + \cdots + 2^{l_p-j_r}\mathbf{z}^+_{x_{jr}} - 2^{l_p-l_1}\mathbf{z}^-_{x_{l_1}} - \cdots - \mathbf{z}^-_{x_{l_p}}\\ &\quad \geq 2^{l_p-j_1}\sum_{k=1}^{p}\mathbf{z}^-_{x_{l_k}} + 2^{l_p-j_2}\mathbf{z}^+_{x_{j_2}} + \ldots + 2^{l_p-j_r}\mathbf{z}^+_{x_{jr}}\\ &\qquad - 2^{l_p-l_1}\mathbf{z}^-_{x_{l_1}} - \cdots - \mathbf{z}^-_{x_{l_p}}\\ &\quad = (2^{l_p-j_1} - 2^{l_p-l_1})\mathbf{z}^-_{x_{l_1}} + \ldots + (2^{l_p-j_1} - 1)\mathbf{z}^-_{x_{l_p}}\\ &\qquad + \underbrace{2^{l_p-j_2}\mathbf{z}^+_{x_{j_2}} + \ldots + 2^{l_p-j_r}\mathbf{z}^+_{x_{jr}}}_{>0} \geq 2^{l_p-j_1}\end{aligned}$$

2. Finally, let us consider the case where it does not exist another index j'' such that $z^-_{x_{j''}} > 0$. As j' is the first index such that $z^+_{x_{j'}} > 0$, then $j' = j_1 < j_2 < \cdots < j_r$ such that $\mathbf{z}^+_{x_{j_k}} > 0$ with $1 \leq k \leq r$, due to the last row of matrix A we have $\mathbf{z}_s = 2^{n-j_1}\mathbf{z}^+_{x_{j_1}} + \cdots + 2^{n-j_r}\mathbf{z}^+_{x_{jr}} \geq 2^{n-j'}\mathbf{z}^+_{x_{j'}}$, and consequently $x_{j'}c^{2^{n-j'}} \leq \mathbf{z}^+$. □

References

Bazgan, C., Hugot, H., & Vanderpooten, D. (2009). Solving efficiently the 0–1 multi-objective knapsack problem. *Computers & Operations Research, 36*, 260–279.

Bertsimas, D., Perakis, G., & Tayur, S. (2000). A new algebraic geometry algorithm for integer programming. *Management Science, 46*(7), 999–1008.

Bertsimas, D., & Weismantel, R. (2005). *Optimization over integers*. Dynamic ideas.

Blanco, V., & Puerto, J. (2009). Partial Gröbner bases for multiobjective integer linear optimization. *SIAM Journal of Discrete Mathemetics, 23*(2), 571–595.

Blanco, V., & Puerto, J. (2011). Some algebraic methods for solving multiobjective polynomial integer programs. *Journal of Symbolic Computations, 46*(5), 511–533.

Boland, N., Charkhgard, H., & Savelsbergh, M. (2014). The triangle splitting method for biobjective integer programming. In *Integer programming and combinatorial optimization*. In *Lecture Notes in Comput. Sci.: 8494* (pp. 162–173). Springer, Cham.

Bringmann, K., Friedrich, T., Igel, C., & Voß, T. (2013). Speeding up many-objective optimization by Monte Carlo approximations. *Artificial Intelligence, 204*, 22–29.

Cox, D. A., Little, J., & O'Shea, D. (2005). Using algebraic geometry. *Graduate texts in mathematics*: 185 (2nd). Springer, New York.

Czyzyk, J., Mesnier, M. P., & Moré, J. J. (1998). The NEOS Server. *IEEE Journal on Computational Science and Engineering, 5*(3), 68–75.

De Loera, J., Hemmecke, R., & Köppe, M. (2013). *Algebraic and geometric ideas in the theory of discrete optimization*. Philadelphia: Society for Industrial and Applied Mathematics.

Dolan, E. D. (2001). The NEOS Server 4.0 administrative guide. *Technical Memorandum ANL/MCS-TM-250*. Mathematics and Computer Science Division, Argonne National Laboratory.

Ehrgott, M. (2005). *Multicriteria optimization* (2nd). Berlin: Springer.

Ehrgott, M., & Gandibleux, X. (2007). Bound sets for biobjective combinatorial optimization problems. *Computers and Operations Research, 34*(9), 2674–2694.

Ehrgott, M., & Ruzika, S. (2008). Improved ϵ-constraint method for multiobjective programming. *Journal of Optimization Theory Applications, 138*(3), 375–396. doi:10.1007/s10957-008-9394-2.

Gropp, W., & Moré, J. J. (1997). Optimization environments and the NEOS server. In M. D. Buhman, & A. Iserles (Eds.), *Approximation theory and optimization* (p. 167). Cambridge University Press.

Kirlik, G., & Sayın, S. (2014). A new algorithm for generating all nondominated solutions of multiobjective discrete optimization problems. *European Journal of Operational Research, 232*, 479–488.

Laumanns, M., Thiele, L., & Zitzler, E. (2006). An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *European Journal of Operational Research, 169*, 932–942.

Mavrotas, G. (2009). Effective implementation of the ϵ-constraint method in multi-objective mathematical programming problems. *Applied Mathematics and Computation, 213*, 455–465.

Mavrotas, G., & Florios, K. (2013). An improved version of the augmented ϵ-constraint method (AUGMECON2) for finding the exact pareto set in multi-objective integer programming problems. *Applied Mathematics and Computation, 219*, 9652–9669.

Mayr, E. W., & Meyer, A. R. (1982). The complexity of the word problems for commutative semigroups and polynomial ideals. *Advance in Mathematics, 46*(3), 305–329.

Natraj, N. R., Tayur, S., & Thomas, R. (1995). An algebraic geometry algorithm for scheduling in presence of setups and correlated demands. *Mathematical Programming, 69*(3, Ser. A), 369–401.

Ralphs, T. K., Saltzman, M. J., & Wiecek, M. M. (2006). An improved algorithm for solving biobjective integer programs. *Annals of Operations Research, 147*(1), 43–70.

Rong, A., & Figueira, J. (2013). A reduction dynamic programming algorithm for the bi-objective integer knapsack problem. *European Journal of Operational Research, 231*, 299–313.

Rong, A., & Figueira, J. (2014). Dynamic programming algorithms for the bi-objective integer knapsack problem. *European Journal of Operational Research, 236*, 85–99.

Schrijver, A. (1986). Theory of linear and integer programming. *Wiley-Interscience Series in Discrete Mathematics*. John Wiley & Sons, Ltd., Chichester. A Wiley-Interscience Publication.

Sturmfels, B. (1996). Gröbner bases and convex polytopes. *University Lecture Series*: 8. Providence, RI: American Mathematical Society.

4ti2 team (2015). 4ti2—a software package for algebraic, geometric and combinatorial problems on linear spaces. Available at www.4ti2.de.

Thomas, R. (1995). A geometric buchberger algorithm for integer programming. *Mathematics of Operations Research, 20*(4), 864–884.

Thomas, R., & Weismantel, R. (1997). Truncated Gröbner bases for integer programming. *Applicable Algebra in Engineering, Communication and Computing, 8*, 241–256.

Zhang, W., & Reimann, M. (2014). A simple augmented ϵ-constraint method for multi-objective mathematical integer programming problems. *European Journal of Operational Research, 234*, 15–24.